

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-145800

(43)Date of publication of application : 06.06.1997

(51)Int.Cl.

G01R 31/3183

G06F 11/22

G06F 17/50

H01L 21/82

(21)Application number : 07-323624

(71)Applicant : NEC CORP

(22)Date of filing : 17.11.1995

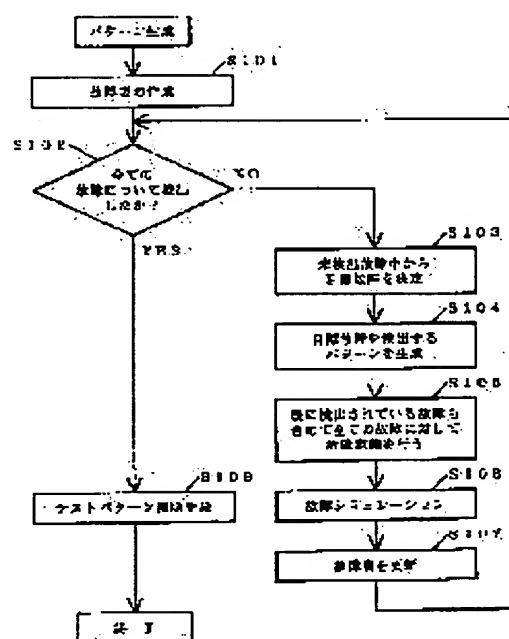
(72)Inventor : MASUMOTO NAGAIRO

(54) TEST PATTERN GENERATION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To reduce test pattern generation time and the number of test patterns when generating the failure detection test pattern of a logic circuit.

SOLUTION: A target failure is selected from a created failure table (S103), a test pattern for detecting the target failure is generated (S104), a failure definition is made to all failures (S105), a failure simulation is performed (S106), a failure table is updated (S107), processing's S103-S107 are repeated until all failures are detected, and a redundant part is deleted from the test pattern (S108). In this case, the test pattern is successively selected from the test patterns with less number of failure detection, it is judged whether all failures detected by the test pattern can also be detected by another test pattern or not, and the test pattern is deleted as a redundant pattern when the failures can also be detected by other test patterns.



LEGAL STATUS

[Date of request for examination] 17.11.1995

[Date of sending the examiner's decision of rejection] 26.01.1999

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection] 11-03182

[Date of requesting appeal against examiner's decision of rejection] 25.02.1999

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-145800

(43)公開日 平成9年(1997)6月6日

(51)Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 1 R 31/3183			C 0 1 R 31/28	Q
G 0 6 F 11/22	3 1 0		C 0 6 F 11/22	3 1 0 B
17/50			15/60	6 7 0
H 0 1 L 21/82			H 0 1 L 21/82	T

審査請求 有 請求項の数6 F D (全 11 頁)

(21)出願番号 特願平7-323624

(22)出願日 平成7年(1995)11月17日

(71)出願人 000004237

日本電気株式会社

東京都港区芝五丁目7番1号

(72)発明者 榎本 長宏

東京都港区芝五丁目7番1号 日本電気株式会社内

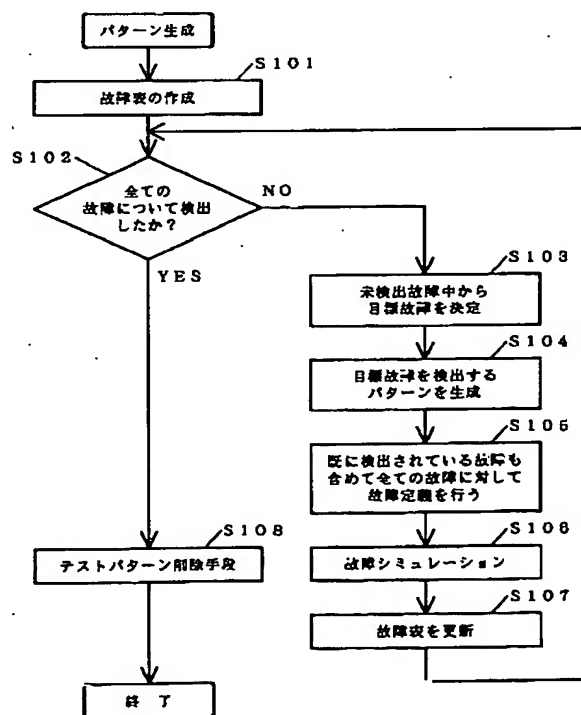
(74)代理人 弁理士 加藤 朝道

(54)【発明の名称】 テストパターン生成方式

(57)【要約】

【課題】論理回路の故障検出テストパターン生成において、テストパターン生成時間及びテストパターン数を小さくする。

【解決手段】作成した故障表から目標故障を選び (S103)、この目標故障を検出するテストパターンを生成し (S104)、全ての故障に対して故障定義を行い (S105)、故障シミュレーションを行い (S106)、故障表を更新し (S107)、処理S103～S107を全ての故障を検出するまで繰り返した後、テストパターンから冗長部分を削除する (S108)。その際、故障検出数の少ない方からテストパターンを順次選び、テストパターンによって検出される全ての故障が他のテストパターンでも検出できるか否かを判定し、他のテストパターンでも検出できる場合にはこのテストパターンを冗長パターンとして削除する。



【特許請求の範囲】

【請求項1】(a)テストパターンの生成対象である回路のうち検出したい故障を選択する工程と、
(b)前記工程(a)で決定した全ての故障のうち目標の故障を選択する工程と、
(c)前記目標の故障を検出するテストパターンを生成する工程と、
(d)前記工程(a)で選択した全ての故障に対して前記工程(c)で生成したテストパターンの故障シミュレーションを行う工程と、
(e)前記テストパターンで検出できない故障を目標の故障とする工程と、
を含み、
前記工程(c)から(e)を繰り返すことにより、前記工程(a)で決定された全ての故障を検出するテストパターンを生成することを特徴とするテストパターン生成方法。

【請求項2】(f)前記全ての故障を検出するテストパターンの中で検出率の低いテストパターンを選択する工程と、
(g)前記検出率の低いテストパターンで検出された全ての故障が他のテストパターンで検出されるか否かを判定する工程と、
(h)前記工程(g)にて他のテストパターンで検出された場合には前記全ての故障を検出するテストパターンの中から前記検出率の低いテストパターンを削除する工程と、
を含み、
前記工程(f)から工程(h)を繰り返すことにより、縮小したテストパターンを生成することを特徴とする請求項1記載のテストパターン生成方法。

【請求項3】(i)前記全ての故障の中で各故障を重複して検出するテストパターンの数が最も少ない故障を選択する工程と、
(j)前記工程(i)にて選択した故障を検出するテストパターンを選択する工程と、
(k)前記工程(j)にて選択したテストパターンで検出されない故障の中で各故障を重複して検出するテストパターンの数が最も少ない故障を選択する工程と、
を含み、
前記工程(j)及び(k)を繰り返すことにより、縮小したテストパターンを生成することを特徴とする請求項1記載のテストパターン生成方法。

【請求項4】テストパターン生成と故障シミュレーションを組み合わせて処理を行うテストパターン生成方式において、
未検出状態の故障の中から目標とする故障を選択し、該目標とする故障を検出するための一又は複数のテストパターンを生成する手段と、
前記生成されたテストパターンと、前記テストパターンのそれぞれによって検出される故障とを対応させて記録する手段と、

前記テストパターンのうちのパターンにより検出される一又は複数の故障が前記テストパターンの他のパターンで検出されるか否かを判定し、該他のパターンで検出された場合には前記一のパターンを削除するテストパターン削除手段と、

を備えたことを特徴とするテストパターン生成方式。

【請求項5】前記テストパターン削除手段が、前記テストパターンを生成する手段により生成されたテストパターン(「リファレンステストパターン」という)のうち故障検出数の少ない方からテストパターンを選択し、該テストパターンによって検出される全ての故障が他のテストパターンによって検出される故障と重複する場合には、該選択された故障検出数の少ないテストパターンを冗長パターンとして前記リファレンステストパターンから削除することを特徴とする請求項4記載のテストパターン生成方式。

【請求項6】前記テストパターン削除手段が、前記テストパターンを生成する手段により生成されたテストパターン(「リファレンステストパターン」という)について故障を重複して検出するテストパターンの数が少ない方から故障を選択し、該選択された故障を検出するテストパターンを必要なものとして残し、前記リファレンステストパターンにより検出される故障についての調査を終了した際に必要とされないテストパターンを冗長パターンとして前記リファレンステストパターンから削除することを特徴とする請求項4記載のテストパターン生成方式。

【発明の詳細な説明】**【0001】**

【発明の属する技術分野】本発明は、論理回路の故障検出方法に関し、特に故障検出テストパターン生成方法に関する。

【0002】

【従来の技術】従来のテストパターン生成方式は、回路図に定義された1つの目標故障を検出するテストパターンを生成する「テストパターン生成」と、あるテストパターンによって検出される故障を求める「故障シミュレーション」と、を有している。従来のテストパターン生成方式において、高い故障検出率を得るテストパターンを生成すると生成されたテストパターン数が長くなる可能性が高く、テスト時間の短縮が困難となるという問題を解消することを目的として、例えば特開平5-341011号公報には、テストパターン生成と故障シミュレーションとを組み合わせて行うテスト生成方法において、1つの故障を目標故障とし、この目標故障を検出するパターンがテストパターン生成によって複数とおり得られた場合、それらのパターンのうちから数本選び、それぞれについて故障シミュレーションを行い、それらの結果のうち、その時点で最も多くの故障を検出できるパターンを目標故障のテストパターンとすることにより、

テストパターン(テストパターン長)の総数を小さくすることを可能とする方法が提案されている。

【0003】次に、上記特開平5-341011号公報に記載の従来のテストパターン生成方式を、図5の流れ図と、テストパターン生成対象の論理回路として図6に示す回路図を用いて説明する。図6には、入力i2、i3を入力とする第1のANDゲートG401、入力i3のインバータG402による反転信号と入力i4とを入力とする第2のANDゲートG403、第1及び第2のANDゲートG401、G403の出力を入力とするORゲートG404、入力i1とORゲートG404の出力とを入力とする第3のANDゲートG405から構成されている。

【0004】まず、ステップS301で故障表(図7参照)を作成する。ステップS302で全ての故障について検出したかを判断し、そうでなければ未検出故障の中から1個の故障目標を取り出し目標故障をする(ステップS303)。最初は、ゲートG405の出力f1の0縮退故障(stuck-at-0、以下「f1/sa0」と記す)を目標故障とする。

【0005】ステップS304で故障f1/sa0が、既に検出されているか否かを判定し、検出されていないならばステップS305で故障f1/sa0を検出するテストパターンを生成する。

【0006】故障f1/sa0が検出されるテストパターンとしては、(i1,i2,i3,i4)=(1,0,0,1)、(1,1,0,1)、(1,1,1,0)、(1,1,1,1)の4本ある。

【0007】生成したテストパターンが複数あるため(ステップS306)、それぞれのテストパターンについて故障シミュレーションを行い(ステップS310)、それぞれの故障検出数を記録する(ステップS311)。

【0008】次に、最も故障検出数の多いテストパターン(1,0,0,1)を選択し(ステップS312)、故障表を更新する(ステップS308)。

【0009】次に、未検出故障の中からゲートG405の出力f1の1縮退故障(stuck-at-1、以下「f1/sa1」と記す)を取り出し、目標故障とする(ステップS303)。ステップS304で故障f1/sa1が、既に検出されているか否かを判定し、検出されていないならばステップS305でf1/sa1を検出するテストパターンを生成する。

【0010】故障f1/sa1が検出されるテストパターンとしては、(i1,i2,i3,i4)=(0,0,0,0)、(0,0,0,1)、(1,0,1,0)、(1,0,1,1)、(1,1,0,0)、(0,0,1,0)、(0,0,1,1)、(0,1,0,0)、(0,1,0,1)、(0,1,1,0)、(0,1,1,1)、(1,0,0,0)の12本ある。生成したテストパターンが複数あるため(ステップS306)、それぞれのテストパターンについて故障シミュレーションを行い(ステップS310)、それぞれの故障検出数を記録する(ステップS311)。

【0011】次に、最も故障検出数の多いテストパターン(1,0,1,1)を選択し(ステップS312)、故障表を更新する(ステップS308)。

【0012】以下、図5の流れ図に基づき、全ての故障が検出されるまで(ステップS302)、上記処理を繰り返す。

【0013】以上の手続により、この従来のテスト生成方法では、図7に示す5本のテストパターン(i1,i2,i3,i4)=(1,0,0,1)、(1,0,1,1)、(0,0,0,1)、(1,1,0,0)により、全てのゲートの入出力f1~f11の0縮退故障(sa0)及び1縮退故障(sa1)を検出することができる。

【0014】

【発明が解決しようとする課題】上述した従来のテストパターン生成方式では、1つの目標故障に対して、複数のテストパターンを生成することが必要とされる他、さらに、複数のテストパターンが得られた場合には、その全てのテストパターンで故障シミュレーションを行わなければならないため、故障シミュレーションの実行に要する時間が膨大になるという問題点を有し、特に半導体集積回路の大規模化と共にこの故障シミュレーションの長大化(すなわち膨大なテストパターン生成時間)は深刻な問題となる。

【0015】従って、本発明は、上記従来技術の問題点に鑑みてなされたものであって、論理回路の故障検出テストパターンの生成において、テストパターン生成時間及びテストパターン数を小さくするテストパターン生成方法及びテストパターン生成方式を提供することを目的とする。

【0016】

【課題を解決するための手段】前記目的を達成するため、本発明は、(a)テストパターンの生成対象である回路のうち検出したい故障を選択する工程と、(b)前記工程(a)で決定した全ての故障のうち目標の故障を選択する工程と、(c)前記目標の故障を検出するテストパターンを生成する工程と、(d)前記工程(a)で選択した全ての故障に対して前記工程(c)で生成したテストパターンの故障シミュレーションを行う工程と、(e)前記テストパターンで検出できない故障を目標の故障とする工程と、を含み、前記工程(c)から(e)を繰り返すことにより、前記工程(a)で決定された全ての故障を検出するテストパターンを生成することを特徴とするテストパターン生成方法を提供する。

【0017】また、本発明は、テストパターン生成と故障シミュレーションを組み合わせて処理を行うテストパターン生成方式において、未検出状態の故障の中から目標とする故障を選択し、該目標とする故障を検出するための一又は複数のテストパターンを生成する手段と、前記生成されたテストパターンと、前記テストパターンのそれぞれによって検出される故障とを対応させて記録す

る手段と、前記テストパターンのうちの一のパターンにより検出される一又は複数の故障が前記テストパターンの他のパターンで検出されるか否かを判定し、該他のパターンで検出された場合には前記一のパターンを削除するテストパターン削除手段と、を備えたことを特徴とするテストパターン生成方式を提供する。

【0018】本発明に係るテストパターン生成方式においては、前記テストパターン削除手段が、前記テストパターンを生成する手段により生成されたテストパターン（「リファレンステストパターン」という）のうち故障検出数の少ない方からテストパターンを選択し、該テストパターンによって検出される全ての故障が他のテストパターンによって検出される故障と重複する場合に、該選択された故障検出数の少ないテストパターンを冗長パターンとして前記リファレンステストパターンから削除することを特徴とする。

【0019】また、本発明に係るテストパターン生成方式においては、前記テストパターン削除手段が、前記テストパターンを生成する手段により生成されたテストパターン（「リファレンステストパターン」という）について故障を重複して検出するテストパターンの数が少ない方から故障を選択し、該選択された故障を検出するテストパターンを必要なものとして残し、前記リファレンステストパターンにより検出される故障についての調査を終了した際に必要とされないテストパターンを冗長パターンとして前記リファレンステストパターンから削除することを特徴とする。

【0020】以下に、本発明の原理・作用を説明する。一般に、故障シミュレーションは非常に多くの時間（シミュレーションの演算時間）がかかる。

【0021】上記構成のもと、本発明は、この故障シミュレーションを少なくすることにより、テストパターン作成を短時間で行えるようにしたものである。本発明は、上記従来のテストパターン生成方法に比べて故障検出率を落とさずにテストパターン生成が行えるという作用効果を有する。すなわち、上記従来のテストパターン生成方法においては、目標故障を検出する全てのパターンを生成し、その各パターンに対して故障シミュレーションをそれぞれ実行し、その中で故障検出率の最も高いパターンを選択していた。この従来の方法では、必然的に故障シミュレーションの回数が多くなる。

【0022】これに対し、本発明は、目標故障を検出するテストパターンをまず1つ生成し、その後未検出故障を検出するパターンを順に1パターンずつ生成する。その上で、パターン単体の故障検出率が低く、他のパターンで代替しても全体の故障検出率に影響を与えないパターンを削除することにより、故障検出率を低下させることなく、テストパターンを縮小化することができる。このため、本発明によれば、テストパターンの生成に要する時間を短縮することができる。

【0023】

【発明の実施の形態】本発明の実施の形態を図面を参照して以下に説明する。

【0024】[実施形態1] 図1及び図2の流れ図を参照して、本発明の一実施形態に係るテストパターン生成方法を以下に説明する。なお、テストパターン生成対象の論理回路としては上記従来例と同様に、図6に示した回路を用いる。

【0025】まず、ステップS101でテストパターン生成を行う回路の故障表（図4参照）を作成する。故障は全てのゲートの入出力に0縮退故障（以下「sa0」と記す）及び1縮退故障（以下「sa1」と記す）を定義する。

【0026】ステップS102で全ての故障について検出したか否かを判定し、全ての故障が検出されていない場合、ステップS103にて未検出故障の中から目標故障を決定して取り出す。

【0027】最初はゲートG405のf1の0縮退故障（「f1/sa0」と記す）を目標故障とし、ステップS104でこの目標故障を検出するためのテストパターンを生成する。

【0028】故障f1/sa0を検出できるテストパターンは、 $(i1, i2, i3, i4) = (1, 0, 0, 1)$ であるので、これをf1/sa0の検出テストパターンとし、既に検出されている故障も含めて全ての故障に対して故障定義を行い（ステップS105）、故障シミュレーションを行う（ステップS106）。

【0029】この検出テストパターンにより、f1/sa0、f2/sa0、f3/sa0、f5/sa0、f8/sa0、f9/sa0、f10/sa0、及びf11/sa1の8個の故障が同時に検出されるため、ステップS107で、テストパターン $(1, 0, 0, 1)$ により検出された故障について故障表の該当する欄に故障検出を表すコードである“det”と記録して故障表を更新する。

【0030】次に、ステップS102で全ての故障について検出したか否かを判定し、この場合、未検出故障が存在するため、ステップS103で未検出故障f1/sa1を選択し、これを目標故障とし、ステップS104でテストパターンを生成する。

【0031】故障f1/sa1を検出できるテストパターンは、 $(i1, i2, i3, i4) = (0, 0, 0, 0)$ であるので、これをf1/sa1の検出テストパターンとし、既に検出されている故障も含めて全ての故障に対して故障定義を行い（ステップS105）、故障シミュレーションを行う（ステップS106）。

【0032】この検出テストパターンにより、f1/sa1の1個の故障が検出されるため、ステップS107でテストパターン $(0, 0, 0, 0)$ により検出された故障について故障表に“det”と記録する。

【0033】以下、図1の流れ図に基づき、全ての故障

が検出されるまで(ステップS102)、上記処理を繰り返す。

【0034】本実施形態では、図4に示した7本のテストパターン($i1, i2, i3, i4 = (1, 0, 0, 1), (0, 0, 0, 0), (0, 0, 0, 1), (1, 0, 1, 0), (1, 1, 1, 0), (1, 1, 0, 0), (1, 0, 1, 1)$)で全てのゲートの入出力の0縮退故障及び1縮退故障を検出することができる。この7本のテストパターンをリファレンスとする。

【0035】次に、上述したリファレンスのテストパターンを用いて、冗長部分を削除する処理であるテストパターン削除手段(ステップS108)の処理行程を図2に示す流れ図と図4の故障表を用いて説明する。

【0036】すなわち、図1のステップS102の判定において、全ての故障について検出した際に、ステップS108にて、上述のテストパターン生成によって生成された7本のテストパターンをリファレンスとして、冗長テストパターンの削除を行う。

【0037】まず、ステップS109でテストパターン生成を行った回路の故障表(図4)を参照する。

【0038】ステップS110で全てのリファレンスのテストパターンについて選択したか否かを判定し、全てのリファレンスのテストパターンについて選択していなければ未選択のテストパターンの中から故障検出数の最も少ないテストパターン(0,0,0,0)を選択する(ステップS111)。

【0039】このテストパターン(0,0,0,0)によって検出される全ての故障(f1/sa1)が他のテストパターンでも検出できるか否かを、故障表(図4参照)を参照して判定する(ステップS112)。

【0040】故障f1/sa1は、テストパターン(0,0,0,1)、(1,0,1,0)、(1,1,0,0)、(1,0,1,1)でも検出できるため、このテストパターン(0,0,0,0)は削除する(ステップS113)。

【0041】次に、ステップS110で全てのテストパターンを選択したか否かを判断し、まだ未選択テストパターンがあるため、未選択テストパターンの中から故障検出数の最も少ないテストパターン(0,0,0,1)を選択する(ステップS111)。

【0042】このテストパターン(0,0,0,1)によって検出される全ての故障(f1/sa1、f2/sa1)が他のテストパターンでも検出できるか否かを故障表(図4参照)を参照して判断する(ステップS112)。

【0043】故障f2/sa1は、他のテストパターンで検出できないため削除できない。

【0044】以下、図2に示す流れ図に基づき、全てのテストパターンが選択されるまで(ステップS110)、上記処理を繰り返す。

【0045】本実施形態では、図4に示すように、テストパターン削除手段により生成されたテストパターンは、($i1, i2, i3, i4 = (1, 0, 0, 1), (0, 0, 0, 1),$

(1,1,1,0)、(1,1,0,0)、(1,0,1,1)の5本となる。

【0046】[実施形態2] 本発明の第2の実施形態として、図1に流れ図を示した前記第1の実施形態によって作成されたリファレンスのテストパターンを用いて、冗長部分を削除する手段の別の形態を、図3の流れ図と図4の故障表を用いて以下に説明する。

【0047】ステップS108で上述のテストパターン生成によって生成された7本のテストパターンをリファレンスとして、冗長テストパターンの削除を行う。

【0048】まず、ステップS201でテストパターン生成を行った回路の故障表(図4)を参照する。

【0049】ステップS202で全ての故障について検出されたか否かを判定し、全ての故障が検出されていない場合、未検出故障の中から、検出しているテストパターン数が最も少ない故障(この場合、1個のテストパターンにより検出される故障f2/sa1)を選択する(ステップS203)。

【0050】この故障f2/sa1を検出しているテストパターン(0,0,0,1)は、必須であるため故障表に記録し(ステップS204)、またこのテストパターン(0,0,0,1)によって検出される他の故障(すなわちf1/sa1)は検出済みとなるため、故障表に記録する(ステップS205)。

【0051】次に、ステップS202で全ての故障について検出されたか否かを判定し、全ての故障について検出していない場合には、未検出故障の中から検出しているテストパターン数が最も少ない故障(この場合、1個のテストパターンにより検出される故障f4/sa0)を選択する(ステップS203)。

【0052】この故障f4/sa0を検出しているテストパターン(1,1,1,0)は、必須であるため故障表に記録し(ステップS204)、またこのテストパターン(1,1,1,0)によって検出される他の故障(f1/sa0、f2/sa0、f3/sa0、f6/sa0、f7/sa0、及びf11/sa0)は、検出済みとなるため故障表に記録する(ステップS205)。

【0053】以下、図3の流れ図に基づき、全ての故障が選択されるまで(ステップS202)、上記処理を繰り返す。

【0054】次に、以上の処理により必須と判定されなかったテストパターンをステップS206において冗長テストパターンとしてリファレンスから削除する。

【0055】本実施形態では、図4に示すように、テストパターン削除手段により生成されたテストパターンは、($i1, i2, i3, i4 = (1, 0, 0, 1), (0, 0, 0, 1), (1, 1, 1, 0), (1, 1, 0, 0), (1, 0, 1, 1)$)の5本となる。

【0056】

【発明の効果】以上説明したように、本発明によれば、テストパターン生成と故障シミュレーションを組み合わせ

せて処理を行うテストパターン生成方式において、故障を検出するための複数のテストパターンを生成する手段と、生成された各々のテストパターンと各テストパターンにより検出される故障を記録する手段と、各テストパターンにより検出される1個以上の故障がその他のテストパターンを削除する手段により、より短時間で故障検出効率の良いテストパターンを生成できる効果がある。

【図面の簡単な説明】

【図1】本発明の一実施形態に係るテストパターン生成方法の処理を説明するための流れ図である。

【図2】本発明の一実施形態におけるテストパターン削除手段の処理を説明するための流れ図である。

【図3】本発明の一実施形態におけるテストパターン削除手段の処理を説明するための流れ図である。

【図4】本発明の一実施形態に係るテストパターン生成

によって作成された故障表を示す図である。

【図5】従来のテストパターン生成方法の処理を説明するための流れ図である。

【図6】テストパターン生成の対象となる回路図である。

【図7】従来のテストパターン生成によって作成される故障表を示す図である。

【符号の説明】

S101～S113、S201～206、S301～S312 ステップ

G401 AND回路

G402 NOT回路

G403 AND回路

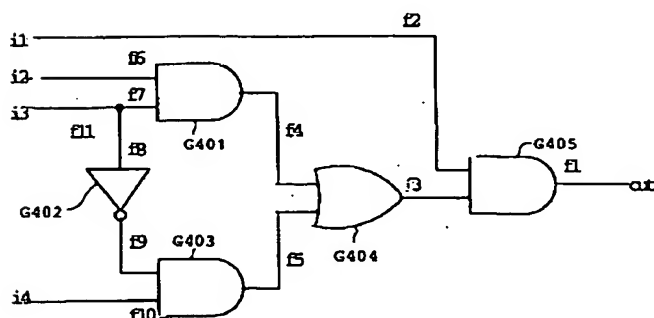
G404 OR回路

G405 AND回路

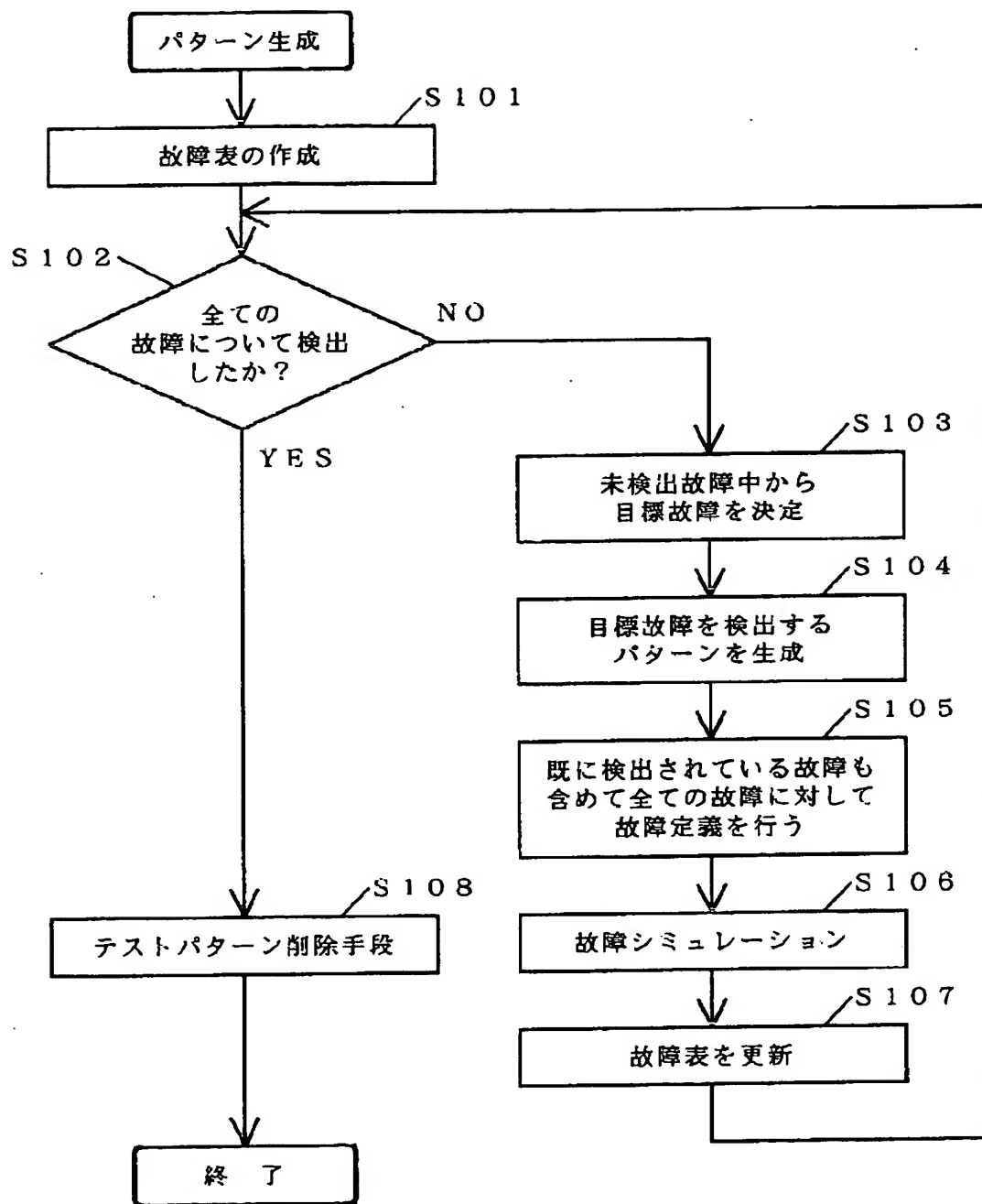
【図4】

パターン	故障検出	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11
1001	f1/ss0	det	det	det			det				det	det
0000	f1/ss0	det										
0001	f2/ss0	det										
1010	f3/ss0	det			det	det	det	det				
1110	f4/ss0	det	det	det	det		det	det				det
1100	f7/ss0	det	det	det	det	det	det		det		det	det
1011	f8/ss0	det			det	det	det		det		det	det

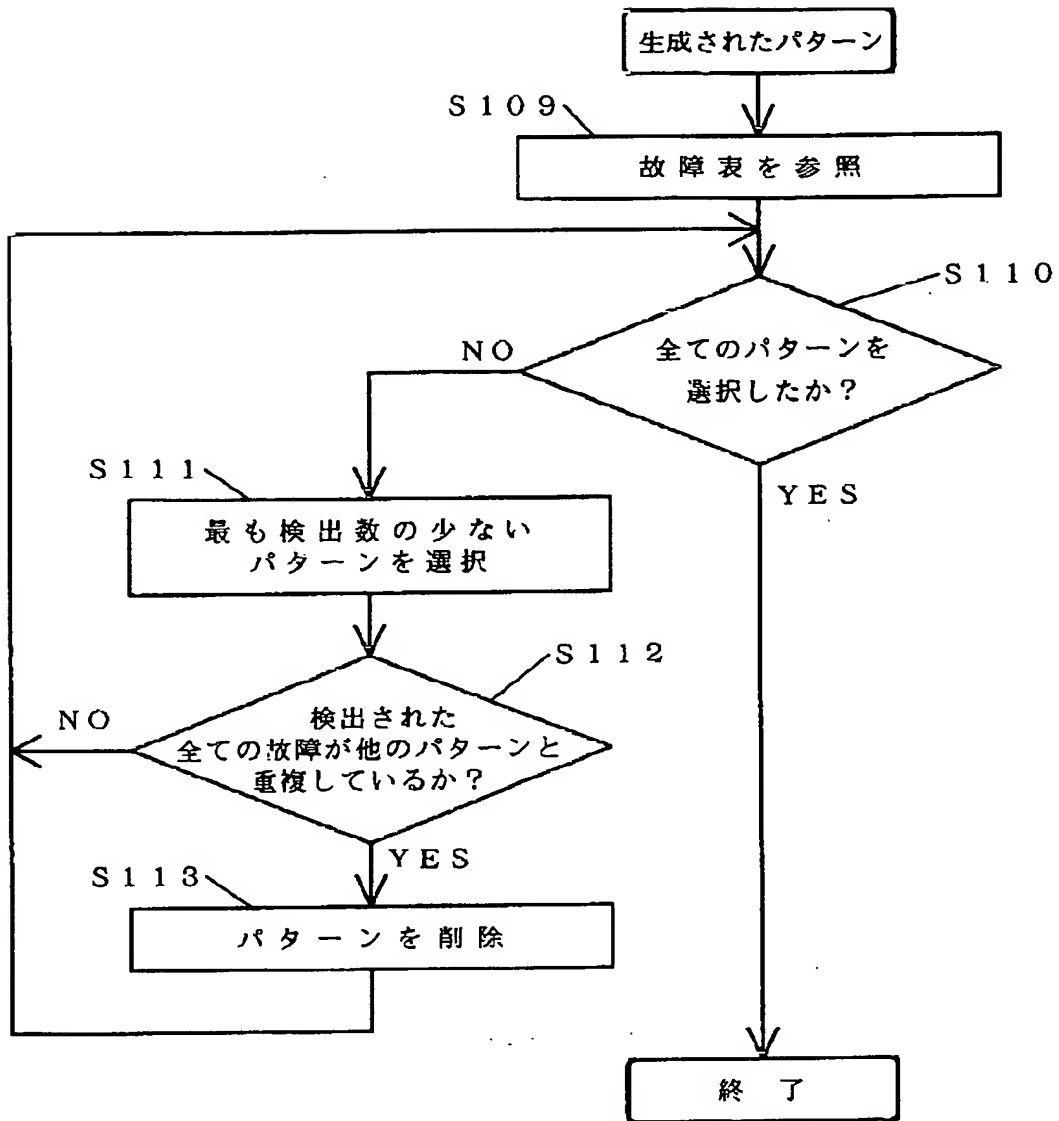
【図6】



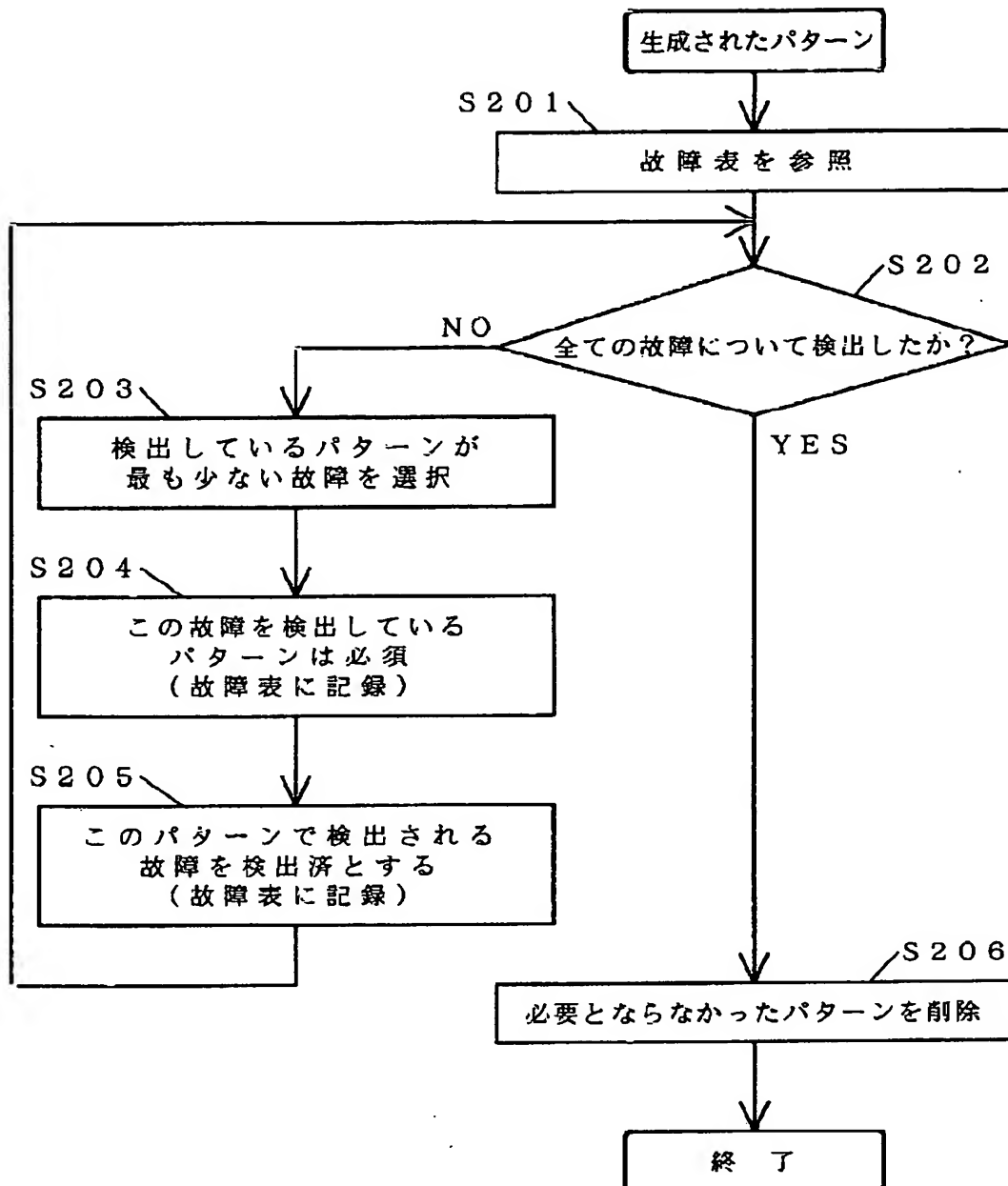
【図1】



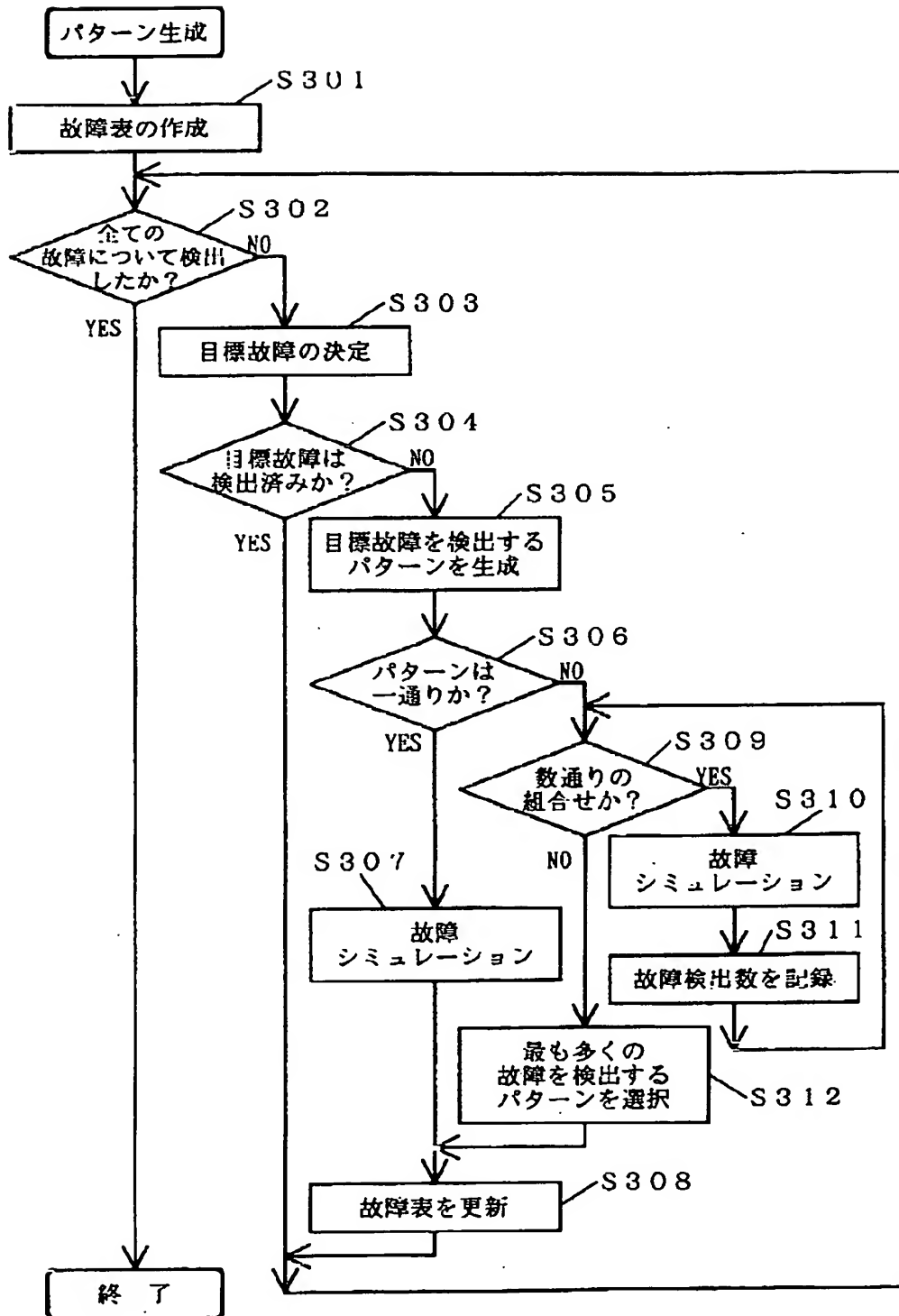
【図2】



【図3】



【図5】



【図7】

目標数値	パターン	1		2		3		4		5		6		7		8		9		10		11	
		ss0	sal	ss0	sal	ss0	sal	ss0	sal	ss0	sal	ss0	sal	ss0	sal	ss0	sal	ss0	sal	ss0	sal	ss0	sal
11/ss0	1001	det		det		det		det		det		det		det		det		det		det		det	
	1101	det		det		det				det						det	det		det				
	1110	det		det		det		det				det		det								det	
	1111	det		det		det		det				det		det									
11/sal	0000	—	det	—		—				—						—	—		—			—	
	0001	—	det	—	det	—				—						—	—		—			—	
	1010	—	det	—		—	det		det		det		det			—	—		—			—	
	1011	—	det	—		—	det		det		det		det		det	—	—	det		det		det	—
	1100	—	det	—		—	det		det		det				det	—	—		—	det		—	
	0010	—	det	—		—				—						—	—		—			—	
	0011	—	det	—		—				—						—	—		—			—	
	0100	—	det	—		—				—						—	—		—			—	
	0101	—	det	—	det	—				—						—	—		—			—	
	0110	—	det	—	det	—				—						—	—		—			—	
	0111	—	det	—	det	—				—						—	—		—			—	
	1000	—	det	—		—	det		det		det					—	—		—	det		—	
12/sal	0001	—	—	—	det	—		—		—		—		—		—	—	—	—	—		—	—
	0101	—	—	—	det	—		—		—		—		—		—	—	—	—	—		—	—
	0110	—	—	—	det	—		—		—		—		—		—	—	—	—	—		—	—
	0111	—	—	—	det	—		—		—		—		—		—	—	—	—	—		—	—
14/ss0	1110	—	—	—	—	—	—	det	—	—	—	det	—	det	—	—	—	—	—	—	—	—	—
	1111	—	—	—	—	—	—	det	—	—	—	det	—	det	—	—	—	—	—	—	—	—	—
17/sal	1100	—	—	—	—	—	—	—	—	—	—	—	—	det	—	—	—	—	—	det	—	—	—